

Programmstruktur

```
# wird einmal bei Programmstart ausgeführt
def setup():
    createCanvas(800,600)

# läuft fortwährend / erzeugt einen Frame
def draw():
    pass # FillMe
```

Systemvariablen

```
# Breite / Höhe des Fensters
windowWidth / windowHeight

# Breite / Höhe des Canvas
width / height

# aktuelle Mausposition
mouseX / mouseY

# Abstand der Maus zum Punkt 0,0
pmouseX, pmouseY

# Nummer des Frames
frameCount
```

Konsolenausgabe

```
print()
```

Farben

Füllfarbe für danach gezeichnete Objekte

```
fill(120)          # grau: 0-255
fill(100,125,255) # r, g, b: 0-255
fill('red')         # Farbbezeichnung
fill('#222222')    # 6-stelliger Farbcode in Hexadezimal
color(0, 0, 255)   # p5 Farbobjekt
```

Hintergrundfarbe + löschen alles Vorherigen

```
background(farbe)
```

Weitere Quellen

<https://learnpython.strivemath.com/p5-python-web>
<https://p5js.org/reference/> (Achtung in JavaScript)

Zufall

```
# kein import nötig (wie sonst in Python)
random(maximum): float
random(minimum, maximum)): float
int(random(minimum, maximum)) # Umwandlung in integer
```

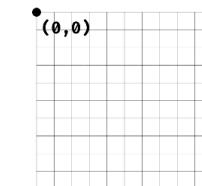
Objekte

```
point(x, y)
line(x1, y1, x2, y2)
circle(x, y, durchmesser)
ellipse(x, y, breite, höhe)
rect(x, y, breite, höhe)
square(x, y, längen)
arc(x, y, breite, höhe, start, stop)

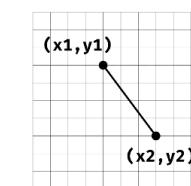
text("text", x, y, breite, höhe)

beginShape()
vertex(x1, y1)
vertex(x2, y2)
vertex(x3, y3)
endShape(CLOSE)
```

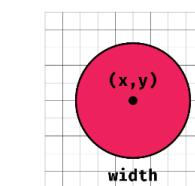
grid system



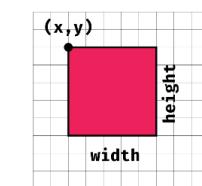
line()



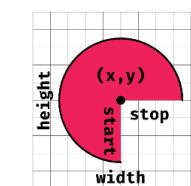
ellipse()



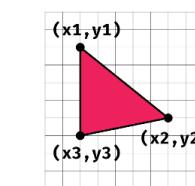
rect()



arc()



vertex()



Methoden

```
# keine Füllfarbe
noFill()

# Strichfarbe
stroke(farbe) | noStroke()

# Strichstärke einstellen
strokeWeight(stärke)

# Schriftgröße
textSize(pixels)

# Bilder
bild = loadImage(URL) | image(bild, x, y)

# Zum selbst recherchieren
linmap(value, start1, stop1, start2, stop2)
bounce(edge1, edge2, speed)
wave(edge1, edge2, speed)

# Positionierung der Objekte
ellipseMode(CENTER) oder ellipseMode(CORNER)
rectMode(CENTER) oder rectMode(CORNER)
coordinateMode(TOP_LEFT) or coordinateMode(BOTTOM_LEFT)
```

Interaktionen

```
# aufgerufen bei Tastendruck
def keyPressed():

# aufgerufen bei Mausklick
def mousePressed():

# ist Taste aktuell gedrückt
# TastenCodes stehen als Variablen zur Verfügung
# UP_ARROW, DOWN_ARROW, LEFT_ARROW..
isKeyDown(keyCode): bool # Bsp: if isKeyDown(UP_ARROW)
```

Weiteres

```
# x- und y- Achse zur Orientierung
drawTickAxes()

# setzt neuen Ursprung, auf den sich alles bezieht
translate(x,y)

# dreht die Zeichenfläche um den Ursprung
rotate(winkel)

# speichert Ursprung und Rotation der Zeichenfläche
push()

# stellt letzten Zustand der Zeichenfläche wieder her
pop()

# berechnet die Distanz zwischen 2 Punkten
dist(x1, y1, x2, y2)
```